UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/575,576	04/11/2006	Reinder J. Bril	BAI525-217/08240	1354

24118 7590 05/11/2010

HEAD, JOHNSON & KACHIGIAN
228 W 17TH PLACE
TULSA, OK 74119

EXAMINER	
KUMABE, BLAKE K	

ART UNIT	PAPER NUMBER
2195	

MAIL DATE	DELIVERY MODE
05/11/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/575,576	BRIL ET AL.	
	Examiner	Art Unit	
	Blake Kumabe	2195	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *06 February 2009*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-24* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-24* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *11 April 2006* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *4/11/2006*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-24 are pending.

Specification

The following guidelines illustrate the preferred layout for the specification of a utility application. These guidelines are suggested for the applicant's use.

Arrangement of the Specification

As provided in 37 CFR 1.77(b), the specification of a utility application should include the following sections in order. Each of the lettered items should appear in upper case, without underlining or bold type, as a section heading. If no text follows the section heading, the phrase "Not Applicable" should follow the section heading:

(a) TITLE OF THE INVENTION.
(b) CROSS-REFERENCE TO RELATED APPLICATIONS.
(c) STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT.
(d) THE NAMES OF THE PARTIES TO A JOINT RESEARCH AGREEMENT.
(e) INCORPORATION-BY-REFERENCE OF MATERIAL SUBMITTED ON A COMPACT DISC.
(f) BACKGROUND OF THE INVENTION.
    (1) Field of the Invention.
    (2) Description of Related Art including information disclosed under 37 CFR 1.97 and 1.98.
(g) BRIEF SUMMARY OF THE INVENTION.
(h) BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWING(S).
(i) DETAILED DESCRIPTION OF THE INVENTION.
(j) CLAIM OR CLAIMS (commencing on a separate sheet).
(k) ABSTRACT OF THE DISCLOSURE (commencing on a separate sheet).
(l) SEQUENCE LISTING (See MPEP § 2424 and 37 CFR 1.821-1.825. A "Sequence Listing" is required on paper if the application discloses a nucleotide or amino acid sequence as defined in 37 CFR 1.821(a) and if the required "Sequence Listing" is not submitted as an electronic document on compact disc).

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

2. Claims 1-12, 14-21, 23-34 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

a. The claim language in the following claims is not clearly understood:

i. As per claim 1, lines 1-2, it is uncertain what the plurality of tasks are scheduled for (i.e. Are the tasks scheduled for execution? Are the tasks scheduled for preemption?). For the purpose of examining, it is construed that tasks are scheduled for preemption.

As per claim 1, lines 3-5, it is uncertain why and how tasks are defined such that matching synchronization primitives does not span a task boundary (i.e. Are matching synchronization primitives methods to reserve and release resources? Are task boundaries points of preemption? Are tasks holding a resource, releasing its resources at a preemption points? Are resources released to avoid deadlock/spinlock?). For the purpose of examining, it is construed that matching synchronization primitives allow tasks to reserve and release resources, task boundaries are points of preemption, and tasks holding

a resource are releasing its resources at a preemption points to avoid deadlock/spinlock.

As per claim 1, lines 6-7, it is unclear what and how a subset of tasks are specified as preemptible or as non-preemptible (i.e. Is a subset of tasks a portion of the plurality of tasks in claim 1 line 1? Are preemptible tasks, tasks that always have to release its resources or tasks may be preempted? Are non-preemptible tasks, tasks that never release its resources or tasks the may never be preempted? Are tasks assigned as preemptible or non-preemptible based on whether or not they are holding resources?). For the purpose of examining, it is construed that tasks of the plurality of tasks are assigned as either preemptible or non-preemptible and preemptability refers to the tasks willingness to release or hold resources.

As per claim 1, lines 8-9, it is uncertain what the suspension data is specifying (i.e. Is suspension equivalent to preemption? Is a preemption point specified by a certain threshold of memory usage? Is the specified preemptability referring to preemptible or non-preemptible in claim 1 line 6?). For the purpose of examining, it is construed that a preemption point is specified as a certain threshold of memory usage and the preemptability is referring to preemptible or non-preemptible in claim 1 line 6.

As per claim 1, lines 10-11, it is uncertain where the input is coming from (i.e. Is a monitoring module measuring the memory usage of each task? Is a task providing its own memory usage to the system?). For the purpose of examining, it is construed that either a monitoring module is measuring the memory usage or each task is providing its won memory usage to the system.

As per claim 1, line 12, it is uncertain how the suspension data is specifying a task is preemptible (i.e. Is the current memory usage compared to a memory usage threshold for preemption? Does the task only need to be capable of preemption?). For the purpose of examining, it is construed that the suspension data is specifying that a current task is at a preemption point based on its memory usage.

As per claim 1, lines 14-15, it is uncertain how synchronization primitives are executed until said suspended tasks terminates (i.e. Are synchronization primitives always enforced? Are resources only held by a task until a task ends or suspends?). For the purpose of examining, it is construed that synchronization primitives are always enforced.

ii. As per claim 12, lines 1-5, it is uncertain why and how tasks are defined such that matching synchronization primitives does not span a task boundary (i.e. Are matching synchronization primitives methods to reserve and release resources? Are task boundaries points of

preemption? Are tasks holding a resource, releasing its resources at a preemption points? Are resources released to avoid deadlock/spinlock?). For the purpose of examining, it is construed that matching synchronization primitives allow tasks to reserve and release resources, task boundaries are points of preemption, and tasks holding a resource are releasing its resources at a preemption points to avoid deadlock/spinlock.

As per claim 12, lines 6-8, it is uncertain where data is being received from and what the data is specifying (i.e. Are tasks supplying the data to the scheduler? Is the maximum memory usage the amount of memory required for a task to execute? Is preemptability based on a tasks willingness to release resources?). For the purpose of examining, it is construed that maximum memory usage us the amount of memory required for a task to execute and preemptability is based on a tasks willingness to release resources.

As per claim 12, lines 8-9, it is unclear what and how a subset of tasks are specified as preemptible or as non-preemptible (i.e. Is a subset of tasks a portion of the plurality of tasks in claim 1 line 1? Are preemptible tasks, tasks that always have to release its resources or tasks may be preempted? Are non-preemptible tasks, tasks that never release its resources or tasks the may never be preempted? Are tasks assigned as preemptible or non-preemptible based on whether or not

they are holding resources?). For the purpose of examining, it is construed that tasks of the plurality of tasks are assigned as either preemptible or non-preemptible and preemptability refers to the tasks willingness to release or hold resources.

As per claim 12, line 12, it is unclear what constitutes a selector (i.e. is the selector equivalent to the scheduler?). For the purpose of examining, it is construed that a selector is equivalent to a scheduler.

As per claim 12 lines 14-15, it is uncertain how synchronization primitives are executed until said suspended tasks terminates (i.e. Are synchronization primitives always enforced? Are resources only held by a task until a task ends or suspends?). For the purpose of examining, it is construed that synchronization primitives are always enforced.

iii. As per claim 15, lines 1-4, it is uncertain why and how tasks are defined such that matching synchronization primitives does not span a task boundary (i.e. Are matching synchronization primitives methods to reserve and release resources? Are task boundaries points of preemption? Are tasks holding a resource, releasing its resources at a preemption points? Are resources released to avoid deadlock/spinlock?). For the purpose of examining, it is construed that matching synchronization primitives allow tasks to reserve and release resources, task boundaries are points of preemption, and tasks holding

a resource are releasing its resources at a preemption points to avoid deadlock/spinlock.

As per claim 15, lines 6-7, it is uncertain where data is being received from and what the data is specifying (i.e. Are tasks supplying the data to the scheduler? Is the maximum memory usage the amount of memory required for a task to execute? Is preemptability based on a tasks willingness to release resources?). For the purpose of examining, it is construed that maximum memory usage us the amount of memory required for a task to execute and preemptability is based on a tasks willingness to release resources.

As per claim 15, line 15, it is unclear what constitutes preemptability (i.e. Is it a tasks willingness to release or hold resources? Are some tasks never allowed to be preempted?). For the purpose of examining, it is construed that preemptability is based on a tasks willingness to release or hold resources.

As per claim 15, lines 16-18, it is uncertain how synchronization primitives are executed until said suspended tasks terminates (i.e. Are synchronization primitives always enforced? Are resources only held by a task until a task ends or suspends?). For the purpose of examining, it is construed that synchronization primitives are always enforced.

iv. As per claim 23, lines 1-6, it is uncertain what the suspension data is specifying (i.e. Is suspension equivalent to preemption? Is a preemption point specified by a certain threshold of memory usage? Is preemptability based on a tasks willingness to release or hold resources? Are some tasks never allowed to be preempted?). For the purpose of examining, it is construed that a preemption point is specified as a certain threshold of memory usage and preemptability is based on a tasks willingness to release or hold resources.

As per claim 23, lines 7-8, it is uncertain why and how tasks are defined such that matching synchronization primitives does not span a task boundary (i.e. Are matching synchronization primitives methods to reserve and release resources? Are task boundaries points of preemption? Are tasks holding a resource, releasing its resources at a preemption points? Are resources released to avoid deadlock/spinlock?). For the purpose of examining, it is construed that matching synchronization primitives allow tasks to reserve and release resources, task boundaries are points of preemption, and tasks holding a resource are releasing its resources at a preemption points to avoid deadlock/spinlock.

As per claim 23, lines 9-10, it is uncertain where the input is coming from (i.e. Is a monitoring module measuring the memory usage of each task? Is a task providing its own memory usage to the system?). For

the purpose of examining, it is construed that either a monitoring module is measuring the memory usage or each task is providing its won memory usage to the system.

As per claim 23, line 11, it is uncertain how the suspension data is specifying a task is preemptible (i.e. Is the current memory usage compared to a memory usage threshold for preemption? Does the task only need to be capable of preemption?). For the purpose of examining, it is construed that the suspension data is specifying that a current task is at a preemption point based on its memory usage.

As per claim 23, lines 14-16, it is uncertain how synchronization primitives are executed until said suspended tasks terminates (i.e. Are synchronization primitives always enforced? Are resources only held by a task until a task ends or suspends?). For the purpose of examining, it is construed that synchronization primitives are always enforced.

v. Claims 17 and 24 have the same deficiency as claim 1.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 1-2, 6, 9, and 17-24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Applicant Admitted Prior Art (AAPA) in view of Sankaranarayan et al. (US 7,284,244) and Kurauchi et al. (US 6,704,489).

4. As per claim 1, AAPA teaches the invention substantially as claimed including a method of scheduling a plurality of tasks in a data processing system (page 1 line 22), comprising the steps of:

for each task of the plurality, providing suspension data specifying suspension of the task based on memory used thereby and on the specified preemptability of the task (page 1 lines 22-28);

processing one of the plurality of tasks (page 1 lines 11-13);

monitoring for an input indicative of memory used by the task matching the suspension data associated with the task (page 2 lines 7-10; page 2 lines 15-17); and

if said suspension data specifies said task is preemptible, performing the steps of:

(i) suspending said task on the basis of said monitored input (page 2 lines 15-19),

(ii) processing a different one of the plurality (page 1 line 22).

AAPA does not specifically teach:

defining each task of said plurality such that a synchronization primitive releasing resources that matches another synchronization primitive protecting resources contained therein does not span a task boundary;

specifying a subset of tasks as preemptible or as non-preemptible depending on whether or not the tasks protect usage of at least one same resource; and

executing synchronization primitives with respect to the protected resources of the suspended task until said suspended task terminates.

However, Sankaranarayan teaches:

defining each task of said plurality such that a synchronization primitive releasing resources that matches another synchronization primitive protecting resources contained therein does not span a task boundary (The resource manager handles the reserving and releasing of resources for tasks. When a first task reserves a shared resource, a second task currently using the shared resource releases the resource before preemption.) (column 8 lines 27-37; column 10 lines 54-56; column 15 lines 29-49); and

executing synchronization primitives with respect to the protected resources of the suspended task until said suspended task terminates (The resource manager arbitrates all resource accesses all the time.) (abstract lines 1-14).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify the tasks taught by AAPA to include primitives that do not span a task boundary taught by Sankaranarayan to allow tasks

shared access to a single resource and prevent deadlock/spinlock due to a task not giving up hold on the resource (column 15 lines 29-49).

The combination of AAPA and Sankaranarayan does not teach specifying a subset of tasks as preemptible or as non-preemptible depending on whether or not the tasks protect usage of at least one same resource. However, Kurauchi teaches specifying a subset of tasks as preemptible or as non-preemptible depending on whether or not the tasks protect usage of at least one same resource (Tasks are assigned an acquisition priority and a use continuation priority. Priorities set if a tasks use of a resource may be preempted or not.) (column 3 lines 25-33; column 15 lines 50-51; column 16 lines 12-45).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify the subset of tasks taught by the combination of AAPA and Sankaranarayan to specify preemptability taught by Kurauchi to give important tasks high priority over a shared resource over lower priority tasks (column 3 lines 25-33; column 15 lines 50-51; column 16 lines 12-45).

5. As per claim 2, AAPA teaches wherein said input comprises data indicative of a suspension request (page 2 lines 7-10).

6. As per claim 6, the combination of AAPA, Sankaranarayan, and Kurauchi teaches the method of claim 1 above. Claim 6 has the same limitations of claim 3 above and is therefore rejected using the same art and rationale as set forth above.

7. As per claim 9, the combination of AAPA, Sankaranarayan, and Kurauchi teaches the method of claim 1 above. Claim 6 has the same limitations of claim 3 above and is therefore rejected using the same art and rationale as set forth above.

8. As per claim 17 AAPA teaches the invention substantially as claimed a method of transmitting data to a data processing system, the method comprising:

transmitting data for use by the data processing system in processing the task; and

transmitting suspension data specifying suspension of the task based on memory usage and preemptability during processing thereof (page 1 lines 22-28), wherein the data processing system is configured to perform a process comprising:

monitoring for an input indicative of memory usage of the task matching the suspension data associated with the task (page 2 lines 7-10; page 2 lines 15-17); and

if said suspension data specifies the task is preemptible, suspending processing of said task on the basis of said monitored input (page 2 lines 15-19).

Sankaranarayan teaches:

defining a task such that a synchronization primitive that protects usage of a resource that matches another synchronization primitive contained therein does not span the task boundary (The resource manager handles the reserving and releasing of resources for tasks. When a first task reserves a shared resource, a second task currently using the shared resource releases the resource before preemption.) (column 8 lines 27-37; column 10 lines 54-56; column 15 lines 29-49); and

executing synchronization primitives with respect to the resources protected by the suspended task until the suspended task terminates (The resource manager arbitrates all resource accesses all the time.) (abstract lines 1-14).

Kurauchi teaches defining all tasks as preemptible or as non-preemptible depending on whether or not the tasks protect usage of at least one same resource (Tasks are assigned an acquisition priority and a use continuation priority. Priorities set if a tasks use of a resource may be preempted or not.) (column 3 lines 25-33; column 15 lines 50-51; column 16 lines 12-45).

9. As per claim 18, AAPA teaches wherein the suspension data includes data identifying maximum memory usage associated with the task (page 1 lines 22-28).

Kurauchi teaches data identifying exclusive resource usage associated with the task and preemptability of the task (Tasks are assigned an acquisition priority and a use

continuation priority. Priorities set if a tasks use of a resource may be preempted or not.) (column 3 lines 25-33; column 15 lines 50-51; column 16 lines 12-45).

10. As per claim 19, AAPA teaches wherein the suspension data identifies at least one point at which processing of the task can be suspended based on memory usage of the task (page 1 lines 22-28).

Kurauchi teaches data identifying exclusive resource usage of the task and preemptability of the task (Tasks are assigned an acquisition priority and a use continuation priority. Priorities set if a tasks use of a resource may be preempted or not.) (column 3 lines 25-33; column 15 lines 50-51; column 16 lines 12-45).

11. As per claim 20, AAPA teaches wherein the task comprises a plurality of sub-jobs and said data identifying at least one point at which processing of the task can be suspended corresponds to each such sub-job that is preemptible (page 1 lines 13-15; page 1lines 22-28).

12. As per claim 21, AAPA teaches wherein the suspension data includes data identifying maximum memory usage associated with the task (page 2 lines 1-6).

Kurauchi teaches data identifying exclusive resource usage of the task (Tasks are assigned an acquisition priority and a use continuation priority. Priorities set if a

tasks use of a resource may be preempted or not.) (column 3 lines 25-33; column 15 lines 50-51; column 16 lines 12-45).

13. As per claim 22, AAPA teaches wherein the task comprises a plurality of sub-jobs and said identifying at least one point at which processing of the task can be suspended corresponds to each such sub-job that is preemptible (page 1 lines 13-15; page 1 lines 22-28).

14. As per claim 23, AAPA teaches the invention substantially as claimed including a method of configuring a task for use in a data processing system (page 1 line 22), the method including associating suspension data with the task, the suspension data specifying suspension of the task based on memory usage associated therewith (page 1 lines 22-28), wherein the data processing system is arranged to perform a process in respect of a plurality of tasks, the process comprising:

monitoring for an input indicative of memory usage of the task matching the suspension data associated with the task (page 2 lines 7-10; page 2 lines 15-17); and

if the suspension data specifies said task is preemptible, suspending processing of said task on the basis of said monitored input (page 2 lines 15-19).

Sankaranarayan teaches:

defining the task such that a synchronization primitive matching another synchronization primitive contain there does not span a task boundary (The resource

manager handles the reserving and releasing of resources for tasks. When a first task reserves a shared resource, a second task currently using the shared resource releases the resource before preemption.) (column 8 lines 27-37; column 10 lines 54-56; column 15 lines 29-49);

executing synchronization primitives with respect to the exclusively used resources of the suspended at least one task until said task terminates (The resource manager arbitrates all resource accesses all the time.) (abstract lines 1-14).

Kurauchi teaches data specifying exclusive resource usage of the task and preemptability of the task (Tasks are assigned an acquisition priority and a use continuation priority. Priorities set if a tasks use of a resource may be preempted or not. Tasks may be assigned to have exclusive resource usage.) (column 3 lines 25-33; column 15 lines 50-51; column 16 lines 12-45).

15. Claim 24 recites a computer program stored in a memory, comprising a set of instructions arranged to cause a processing system to perform the above steps. It has the same limitations of claim 1 above and is therefore rejected using the same art and rationale as set forth above.

16. Claims 3-5, 7-8, and 10-11 are rejected under 35 U.S.C. 103(a) as being unpatentable over AAPA, Sankaranarayan, and Kurauchi as applied to claims 2, 6, and 9, respectively, above, and further in view of Bishop et al. (US 5,826,082).

17. As per claim 3, AAPA does not specifically teach the steps of:

receiving first data identifying maximum memory usage associated with the plurality of tasks;

receiving second data identifying memory available for processing the plurality of tasks; and

identifying, on the basis of the fast and second data, whether there is sufficient memory available to process the tasks;

wherein, said monitoring, suspending steps, and executing steps are performed only in response to identifying insufficient memory.

However, Bishop teaches:

receiving first data identifying maximum memory usage associated with the plurality of tasks (column 3 lines 55-57);

receiving second data identifying memory available for processing the plurality of tasks (column 3 lines 64-67); and

identifying, on the basis of the fast and second data, whether there is sufficient memory available to process the tasks (column 3 lines 64-67);

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify the method taught by the combination of AAPA, Sankaranarayan, and Kurauchi to include the steps taught by Bishop to

accommodate the uncertainty of whether resources needed to complete an operation will be available (column 1 lines 13-14).

The combination of AAPA, Sankaranarayan, Kurauchi, and Bishop does not specifically teach wherein, said monitoring, suspending steps, and executing steps are performed only in response to identifying insufficient memory. However, it would have been obvious to a person of ordinary skill in the art at the time the invention was made not to perform the suspending step because there is sufficient memory for the task. As such, it would also have been obvious not to perform the monitoring step because doing so would waste resources monitoring for a suspension event that has already been determined not to occur.

18. As per claim 4, Bishop teaches the steps of:

monitoring termination of tasks (column 5 lines 64-65); and

in response to a task termination, repeating said step of identifying availability of memory in response to a task terminating (Once the first thread finishes, there is sufficient memory to unsuspend the second request/thread.) (column 5 lines 64-67).

19. As per claim 5, the combination of AAPA, Sankaranarayan, Kurauchi, and Bishop does not specifically teach in response to identifying sufficient memory to execute the remaining tasks, the monitoring step is deemed unnecessary. However, it would have been obvious to a person of ordinary skill in the art at the time the invention was made

that the monitoring step be deemed unnecessary because performing the step would waste resources monitoring for a suspension event that has already been determined not to occur, since there is sufficient memory for the task.

20. As per claims 7-8, the combination of AAPA, Sankaranarayan, and Kurauchi teaches the method of claim 1 above. Claims 7-8 have the same limitations of claims 4-5, respectively, above and is therefore rejected using the same art and rationale as set forth above.

21. As per claims 10-11, the combination of AAPA, Sankaranarayan, and Kurauchi teaches the method of claim 1 above. Claims 10-11 have the same limitations of claims 4-5, respectively, above and is therefore rejected using the same art and rationale as set forth above.

22. Claims 12-16 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bishop et al. (US 5,826,082) in view of Sankaranarayan et al. (US 7,284,244) and Kurauchi et al. (US 6,704,489).

23. As per claim 12, Bishop teaches the invention substantially as claimed including a scheduler for use in a data processing system, the data processing system being arranged to execute a plurality of tasks and having access to a specified amount of

memory (127) for use in executing the tasks (abstract; Figure 1; column 2 lines 29-34; column 3 lines 5-17), the scheduler comprising:

a data receiver arranged to receive data identifying maximum memory usage associated with a task (column 3 lines 55-57),

an evaluator arranged to identify, on the basis of the received data, whether there is sufficient memory to execute the tasks (column 3 lines 64-67); and

a selector arranged to select at least one task for suspension during execution of the task, said suspension coinciding with a specified memory usage by the task and the task being preemptible (column 4 lines 21-23; column 5 lines 5-11; column 5 line 28);

wherein, in response to the evaluator identifying that there is insufficient memory to execute the plurality of tasks,

- the selector selects at least one task for suspension, on the basis of its specified memory usage, and the specified amount of memory available to the data processing system, and
- the scheduler suspends execution of the at least one selected task in response to the task using the specified memory (column 4 lines 21-23; column 5 lines 5-11; column 5 line 28).

Bishop does not specifically teach:

tasks defined such that a synchronization primitive releasing resources matching another synchronization primitive protecting resources contained therein does not span a task boundary;

data identifying exclusive resource usage of the task, and preemptability of the task, wherein a subset of said plurality of tasks protecting usage of the same resource are all identified as one of preemptible or non-preemptible; and

execution thereafter of synchronization primitives with respect to the protected resources of the suspended at least one task until said suspended at least one task terminates.

However, Sankaranarayan teaches:

tasks defined such that a synchronization primitive releasing resources matching another synchronization primitive protecting resources contained therein does not span a task boundary (The resource manager handles the reserving and releasing of resources for tasks. When a first task reserves a shared resource, a second task currently using the shared resource releases the resource before preemption.) (column 8 lines 27-37; column 10 lines 54-56; column 15 lines 29-49); and

execution thereafter of synchronization primitives with respect to the protected resources of the suspended at least one task until said suspended at least one task terminates (The resource manager arbitrates all resource accesses all the time.) (abstract lines 1-14).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify the tasks taught by Bishop to include primitives that do not span a task boundary taught by Sankaranarayan to allow tasks

shared access to a single resource and prevent deadlock/spinlock due to a task not giving up hold on the resource (column 15 lines 29-49).

The combination of Bishop and Sankaranarayan does not specifically teach data identifying exclusive resource usage of the task, and preemptability of the task, wherein a subset of said plurality of tasks protecting usage of the same resource are all identified as one of preemptible or non-preemptible. However, Kurauchi teaches data identifying exclusive resource usage of the task, and preemptability of the task, wherein a subset of said plurality of tasks protecting usage of the same resource are all identified as one of preemptible or non-preemptible (Tasks are assigned an acquisition priority and a use continuation priority. Priorities set if a tasks use of a resource may be preempted or not.) (column 3 lines 25-33; column 15 lines 50-51; column 16 lines 12-45).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify the subset of tasks taught by the combination of Bishop and Sankaranarayan to specify preemptability taught by Kurauchi to give important tasks high priority over a shared resource over lower priority tasks (column 3 lines 25-33; column 15 lines 50-51; column 16 lines 12-45).

24. As per claim 13, Bishop teaches monitoring termination of tasks, and in response to a task terminating, to identify whether there is sufficient memory to execute the remaining tasks (column 5 lines 64-67).

25. As per claim 14, Bishop teaches in response to the evaluator identifying sufficient memory to execute the remaining tasks the selector is arranged to deselect said selected at least one task (column 5 lines 64-67).

26. As per claim 15, Bishop teaches the invention substantially as claimed including a data processing system arranged to execute a plurality of tasks (abstract; Figure 1; column 2 lines 29-34; column 3 lines 5-17), the data processing system including:

memory arranged to hold instructions and data during execution of a task (104) (Figure 1);

receiving means arranged to receive data identifying maximum memory usage associated with a task (column 3 lines 55-57);

evaluating means arranged to identify, on the basis of the received data, whether there is sufficient memory to execute the tasks (column 3 lines 64-67);

a scheduler arranged to schedule execution of the tasks on the basis of input received from the evaluating means (column 4 lines 21-23),

wherein, in response to identification of insufficient memory to execute the plurality of tasks, the scheduler is arranged to suspend execution of at least one task in

dependence on memory usage by the task (column 4 lines 21-23; column 5 lines 5-11; column 5 line 28).

Sankaranarayan teaches:

task defined such that a synchronization primitive matching another synchronization primitive contained therein does not span a task boundary (The resource manager handles the reserving and releasing of resources for tasks. When a first task reserves a shared resource, a second task currently using the shared resource releases the resource before preemption.) (column 8 lines 27-37; column 10 lines 54-56; column 15 lines 29-49); and

to direct the execution thereafter of synchronization primitives with respect to the protected resources of the suspended at least one task until said suspended task terminates (The resource manager arbitrates all resource accesses all the time.) (abstract lines 1-14).

Kurauchi teaches:

data specifying preemptability of the task; and

suspending execution of at least one task in dependence on exclusive resource usage by the task, and preemptability of the task (Tasks are assigned an acquisition priority and a use continuation priority. Priorities set if a tasks use of a resource may be preempted or not. Tasks may be assigned to have exclusive resource usage.) (column 3 lines 25-33; column 15 lines 50-51; column 16 lines 12-45).

27. As per claim 16, Kurauchi teaches wherein a subset of said plurality of tasks is determined be preemptible or non-preemptible depending on whether or not the subset of tasks protect usage of the same resource (Tasks are assigned an acquisition priority and a use continuation priority. Priorities set if a tasks use of a resource may be preempted or not. When a resource request is received the acquisition and continuation priorities of the requesting and the current task are compared to determine if the current task should be preempted.) (column 3 lines 25-33; column 15 lines 50-51; column 16 lines 12-45).

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Narlikar ("Space-Efficient Implementation of Nested Parallelism", 7/1997, ACM, pages 25-36) discloses a space efficient implementation of nested parallelism.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Blake Kumabe whose telephone number is 571-270-5593. The examiner can normally be reached on 7:30am - 5:00pm EST Monday to Thursday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on 571-272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Meng-Ai An/
Supervisory Patent Examiner, Art Unit 2195

/B. K./
Examiner, Art Unit 2195